

Policy Optimization for Content Push via Energy Harvesting Small Cells in Heterogeneous Networks

Jie Gong *Member, IEEE*, Sheng Zhou *Member, IEEE*,
Zhenyu Zhou *Member, IEEE*, Zhisheng Niu *Fellow, IEEE*

Abstract

Motivated by the rapid development of energy harvesting technology and content-aware communication in access networks, this paper considers the push mechanism design in small-cell base stations (SBSs) powered by renewable energy. A user request can be satisfied by either push or unicast from the SBS. If the SBS cannot handle the request, the user is blocked by the SBS and is served by the macro-cell BS (MBS) instead, which typically consumes more energy. We aim to minimize the ratio of user requests blocked by the SBS. With finite battery capacity, Markov decision process based problem is formulated, and the optimal policy is found by dynamic programming (DP). Two threshold-based policies are proposed: the push-only threshold-based (POTB) policy and the energy-efficient threshold-based (EETB) policy, and the closed-form blocking probabilities with infinite battery capacity are derived. Numerical results show that the proposed policies outperform the conventional non-push policy if the content popularity changes slowly or the content request generating rate is high, and can achieve the performance of the greedy optimal threshold-based (GOTB) policy. In addition, the performance gap

Jie Gong is with School of Data and Computer Science, Sun Yat-sen University, Guangzhou 510006, P. R. China. Email: gongj26@mail.sysu.edu.cn.

Sheng Zhou and Zhisheng Niu are with Tsinghua National Laboratory for Information Science and Technology, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China. Emails: {sheng.zhou, niuzhs}@tsinghua.edu.cn.

Zhenyu Zhou is with State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, School of Electrical and Electronic Engineering, North China Electric Power University, Beijing 102206, China. Email: zhenyu_zhou@ncepu.edu.cn

This work is sponsored in part by the Fundamental Research Funds for the Central Universities, the National Basic Research Program of China (973 Program: No.2012CB316001), the Nature Science Foundation of China (61571265, 61321061, 61601180, 61601181, and 61461136004) and Hitachi R&D Headquarter.

between the threshold-based policies and the DP optimal policy is small when the energy arrival rate is low or the request generating rate is high.

I. INTRODUCTION

With the rapid growth of the wireless multimedia traffic, the enormous energy consumption of wireless communication systems has become one of the major concerns in the future 5G networks. Energy harvesting (EH) technology [1]–[7], which utilizes the renewable energy from natural sources such as solar, wind, kinetic activities and so on, can greatly reduce the power consumption from the conventional grid power supply, i.e., power grid, and hence reduce the CO₂ emissions. It has been considered as one of the candidate technologies to achieve green communications. On the other hand, the heterogeneous wireless network, where the small-cell base stations (SBSs) with low transmit power take in charge of a small area to enhance the overall performance, is considered as a possible future network architecture. The SBSs can be powered by renewable energy to reduce deployment and operation costs [8], [9]. During the periods of energy deficiency, the users can instead be served by the macro-cell base stations (MBSs) with grid power supply. However, due to the randomness of the renewable energy arrival and the limitation on the battery capacity, energy shortage or waste will occur when the traffic pattern mismatches the energy harvesting process in either spatial or time domain. Content caching and push [10]–[21] at access points and end users can effectively reduce the duplicated content transmissions, and hence, reduce the core network traffic load as well as the energy consumption. As a result, it is promising to deal with the mismatch between energy and traffic via content caching and push in renewable energy powered small cells.

In the literature, there have been extensive studies on how to match harvested energy with traffic profile. The stability condition and the average delay of the queuing system with both energy queue and data queue are analyzed in [2]. The work in [3] studies the optimal power allocation for transmission time minimization with random packet arrival under additive white Gaussian noise (AWGN) channel. The power allocation problem is further extended to the fading channel case in [4], [5]. The long-term spatial and temporal variations of mobile traffic are considered in [6], [7], and the renewable energy is adapted through resource allocation and base station (BS) sleeping, respectively. The small BS on/off scheduling in heterogeneous networks is also studied in [22] to minimize energy consumption and transmission delay. There are other

ways to adjust the energy profile between nodes, for example, devices can trade the harvested energy with one another based on the service requirement [23], and the secondary user in cognitive radio networks can harvest the RF energy from the primary user and use it to transmit in idle times [24]. The existing studies mainly focus on the algorithm design based on the traffic intensity, i.e., how many bits or packets need to be transmitted. However, the important content information carried inside the bit stream has not been well explored.

In this paper, we try to further improve the performance by exploring the content information. Our work can be viewed as part of the recently proposed GreenDelivery framework [25], which adopts content caching and push in EH powered SBSs to enable efficient content delivery. Statistic data shows that people's interests to the contents are not equal [26]. Wireless multicast [27], [28] holds the promise of achieving huge energy efficiency gain via broadcasting commonly interested multimedia contents to multiple users simultaneously. Hence, the content push mechanism [10] is developed based on multicast to make full use of the content popularity features. A learning algorithm for adapting the proactive push to the dynamic client demands is proposed in [11]. And the capacity gain by push in the integrated broadcast and communication network is analyzed in [12]. In reality, proactive push is supported by the development of last-mile wireless access hardwares and mobile devices. To reduce the core network overhead, contents are suggested to be cached at the SBSs [13]–[17] or relay nodes [18], with proactive caching schemes [19]–[21]. The work in [29] uses multicast to satisfy users' common requests in the cache-enabled heterogeneous cellular networks and formulates the problem by Markov decision process (MDP) [30]. However, pushing contents prior to the actual requests is not considered. In addition, the dynamic energy arrival is not considered either. Recently, EH based SBSs are used to cache contents [31] for improving deployment flexibility and reducing energy consumption. In [32], an online power control scheme is developed for EH based SBS with wireless backhaul and local cache. On the other hand, with the rapid improvement of data storage capacity, user devices are capable of storing large amount of data. With the available contents at the SBSs and the large user storage capacity, proactive push by EH powered SBSs in heterogeneous cellular networks is considered to be practical and effective.

Combining EH with content push provides potentials to improve the efficiency of the renewable energy. Specifically, the contents are pushed to users earlier than the actual demands when the harvested energy is sufficient. In reward, the energy waste due to the battery overflow can be

avoided as the harvested energy can be used effectively and timely. Later, when the energy state becomes poor, the content requests can be satisfied by users' local storage. It can be regarded as *transferring* the harvested energy along with the timeline to the *future* to match the content requests. However, the push mechanism design in EH powered SBSs is still an open problem, which is not trivial since it needs to jointly consider energy state, traffic load as well as content popularity. The advantage of content push is that it reduces the duplicated transmissions of the same content. But at the same time, pushing a content typically consumes more energy than unicast as it needs to guarantee the data rate of the worst-channel user. Hence, there is a tradeoff between the energy saving due to the reduction of the duplicated content transmissions and the energy consumption of content push. We try to explore the tradeoff and design efficient push mechanisms. Notice that if the user request cannot be satisfied in the small-cell and has to be handled by the MBS, we say that it is blocked by the SBS. The objective is to minimize the blocking probability in the SBS. The preliminary optimal policy design has been presented in [33]. This paper extensively considers the design of policies with low-complexity. The contributions are as follows:

- With finite battery capacity, we formulate the problem using MDP tool by discretizing the system state (including the battery state, the content request and the set of contents which have been pushed) into a finite set, and find the optimal stationary policy via dynamic programming (DP).
- We propose two threshold-based policies: The push-only threshold-based (POTB) policy and the energy-efficient threshold-based (EETB) policy. With infinite battery capacity, we derive the closed-form blocking probability which equals to the result under average power constraint. In finite battery capacity, the threshold-based policies can be analyzed via the finite state Markov chain (FSMC) model.
- The performances of the proposed policies are evaluated via numerical simulations compared with the conventional non-push policy, the greedy optimal threshold-based (GOTB) policy and the DP optimal policy. The sensitivity of the proposed threshold-based policies to the battery capacity is also studied.

The rest of the paper is organized as follows. Section II presents the system model and the problem formulation. Section III provides the MDP analysis and the policy iteration algorithm to find the optimal policy. The threshold-based policies are proposed and analyzed in Section

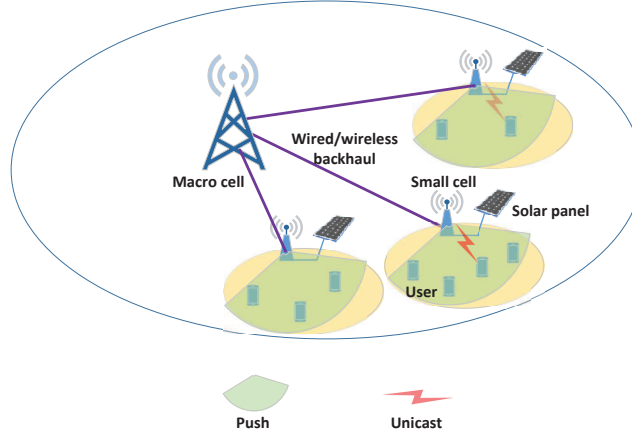


Fig. 1. Two-tier heterogeneous cellular network. SBSs can fetch contents via backhaul link and store in their cache. The cached contents can be either pushed to all users or unicasted to one of them.

IV. Numerical results are provided in Section V for performance evaluation. Finally, Section VI concludes the paper.

II. SYSTEM MODEL AND PROBLEM FORMULATION

We consider a two-tier heterogeneous cellular network composed of a MBS and multiple SBSs as shown in Fig. 1. Each SBS serves the users in its coverage and treats the data transmissions of other SBSs as background interference. Since the SBSs are densely deployed, the sum interference is considered the same for each user. Hence, we only focus on the policy design of a single SBS and all the SBSs can work in the same way. The MBS is powered by the power grid and the SBS is powered by renewable energy solely. The harvested energy is stored in a battery with finite capacity. The SBS is connected with the MBS via a wired/wireless backhaul link to fetch contents and store them in its cache to reduce the congestion of core networks. We assume in this paper the time and energy consumption to fetch contents is negligible, and thus only focus on how to push contents from the SBS to users.

Fig. 2 illustrates how the dynamic system evolves. Specifically, the system is slotted with slot length T_p . In each slot k , a content request Q_k is generated from the user side. Based on the request and the battery energy state, the SBS takes an action with energy usage U_k , and then harvests a certain amount of energy A_k . At the user side, each user receives the contents and preserves a list of pushed contents C_k . The detailed description of the model is as follows.

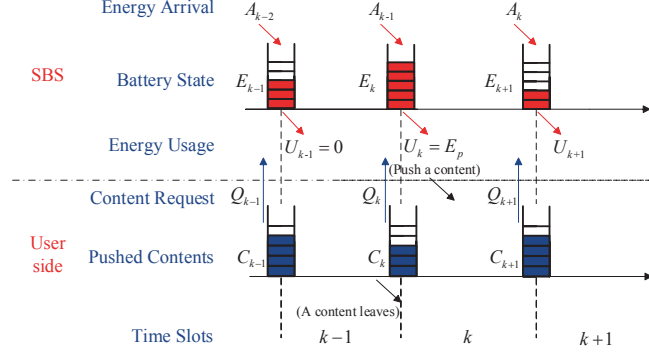


Fig. 2. Timeline of the slotted system and examples of actions. At the beginning of slot $k - 1$, the SBS takes sleep action ($U_{k-1} = 0$). By the end of slot $k - 1$, a content leaves the system, and the number of pushed contents C_k is reduced by 1. At the beginning of slot k , the SBS takes push action ($U_{k-1} = E_p$), which results in an increase of C_{k+1} by the end of the slot.

A. Content Request and Channel Model

The content request is assumed to follow the Bernoulli distribution, i.e., there is a content request with probability $p_u \in [0, 1]$ in each slot. And the location of the request follows the uniform distribution in the cell coverage. The reasons for using Bernoulli distribution are as follows. Firstly, due to the limited energy availability, the renewable energy powered SBS is usually deployed with small coverage or in regions of low traffic load, i.e., in most cases, there is no more than one request at one time. If multiple contents are requested simultaneously, they are probably the popular ones which can be satisfied by the users' local storage and will not trigger simultaneous multiple transmissions. As a consequence, our assumption is a good approximation of the real system. Secondly, the total number of content requests in the macro-cell (summation of independent Bernoulli distributed random variables) follow the binomial distribution, which can be well fitted by a Poisson distribution as the SBSs are densely deployed. Hence, the model for generating content request in macro-cell well meets the typical Independent Reference Model (IRM) provided in [34].

Each content is transmitted in one time slot with fixed target average data rate r_0 . For each content transmission, the average data rate of a given user can be calculated as

$$r_0 = \mathbb{E}_h \left[W \log_2 \left(1 + \frac{P_t |h|^2 \beta d^{-\alpha}}{\sigma^2 + P_I} \right) \right], \quad (1)$$

where W is the bandwidth of the SBS, P_t is the transmit power, h is the small-scale fast fading

coefficient, β and α represent the pathloss constant and the pathloss exponent, respectively, d is the transmission distance, $\sigma^2 + P_I$ is the noise plus interference power. Assume the SBSs and the MBS are allocated with orthogonal frequency bands, respectively. As a result, the interference is only caused by the SBSs working in the same frequency band. And the interference power P_I is computed by the summation over all the interfering SBSs. \mathbb{E}_h is the expectation operator with respect to h . That is, the average data rate r_0 is obtained by averaging the instantaneous channel rates of all fading blocks. In practice, as a content usually contains multiple data packets, each transmission slot contains multiple packet transmission frames, among which the small scale fading h changes. Hence, the data rate averaging over h is a good approximation of the actual content delivery rate. We also assume that the user is of low mobility so that it remains in one cell during transmission. Based on the channel model, the required transmit power P_t can be obtained by solving (1) numerically. This can be done by the bi-section method [35, Chap. 2.1] for instance. To calculate the rate, the channel state information (CSI) is required. In real networks, the users will get the CSI via standard channel training process. In addition, only a small number of bits will be fed back to the SBS for power control.

Based on the above model, the content request state Q_k can be denoted as the amount of energy required for unicasting a content. As the calculation is averaged over h , Q_k is a function of the transmission distance, i.e., $Q_k = E_u(d)$. Notice that $Q_k = 0$ indicates that there is no unicast request.

B. Action and Energy Model of the SBS

The SBS has three possible actions: sleep, unicast, and push. When the SBS has sufficient energy, it can either unicast a content upon request, or push a popular content to all the users in its coverage. When the battery energy is depleted, the SBS enters the sleep mode and the content request will be handled by the MBS. The action is denoted by u_k and is defined as

$$u_k = \begin{cases} 0, & \text{sleep} \\ 1, & \text{unicast a content} \\ 2, & \text{push a content} \end{cases} \quad (2)$$

Notice that when push action is taken, the SBS always pushes the most popular content to the users. This is the most efficient as it minimizes the number of the duplicated data transmissions.

The action to be taken is constrained by the available battery energy, i.e., the energy used in slot k satisfies $U_k \leq E_k$. In sleep action, we assume the energy consumption is negligible, i.e., $U_k = 0$. In push action, the energy $U_k = E_p$ is used to guarantee that all the users in the cell coverage can successfully receive the content. Then the battery energy state is updated as $E_{k+1} = \min\{B_{\max}, E_k - U_k + A_k\}$, where B_{\max} is the battery capacity, and the harvested energy A_k is assumed ergodic and i.i.d. with mean \bar{A} .

C. Content Popularity Model

Assume there are in total N contents that the users are interested in, denoted by the list $\mathbf{C} = (c_1, \dots, c_N)^T$, where the i -th ranked content c_i is requested with probability (or popularity) f_i . Statistical researches have shown that the content popularity distribution is well fitted by the Zipf distribution [13], [26]. The popularity of content c_i can be expressed as

$$f_i = \frac{1/i^v}{\sum_{j=1}^N 1/j^v}, \quad (3)$$

where $v \geq 0$ is the skew parameter. Obviously, we have $f_1 > \dots > f_N$.

In addition, as people's interest changes over time, old contents may be outdated and new contents will attract people's interest. We assume in each slot there is an old content replaced by a new one with probability $p_c \in [0, 1]$, and the old one is randomly picked over \mathbf{C} with equal probability $\frac{1}{N}$. While the popularity changes as follows. If content c_i is replaced by the new content c' , the content list becomes $\mathbf{C}' = (c_1, \dots, c_{i-1}, c_{i+1}, \dots, c_N, c')^T$, i.e., the ranks of the contents c_{i+1}, \dots, c_N increase by one and the new content c' is with rank N . Such a content updating model is used for two reasons. Firstly, it reflects the fact that the popularity of most contents increases gradually in real systems [26]. Secondly, it simplifies the problem formulation, i.e., we only need to consider the number of popular contents rather than the popularity of each individual content. Specifically, according to the push action and content model, it can be easily verified that the pushed content list is $\mathbf{C}_k = (c_1, \dots, c_{C_k})$, where $C_k \leq N$ is the number of pushed contents. That is, the users always preserve the most popular contents. The push action is taken on the remained list $\tilde{\mathbf{C}}_k = (c_{C_k+1}, \dots, c_N)$, and the content c_{C_k+1} should be pushed at first. As a result, we can use the scalar C_k instead of the vector \mathbf{C}_k to indicate the push state.

In summary, the randomness involved in the system model includes the following. Each content request is generated randomly according to Bernoulli distribution. Which content is requested

follows the popularity distribution defined in (3), and its location follows uniform distribution. The random location of the user request indicates the randomness of channel. In addition, the energy arrives randomly, and the content list is updated randomly. As the system state changes among slots, the action should also change accordingly.

D. Problem Formulation

When a user requests a content that is not in its storage, and at the same time, the SBS does not unicast the content, the request is blocked by the SBS and needs to be handled by the MBS, which usually causes more energy consumption due to the larger transmission distance. We aim at optimizing the usage of harvested energy to minimize the ratio of content requests blocked by the SBS, which is named as the *blocking probability* at the SBS in the rest of the paper. Mathematically, the objective is expressed as

$$\underset{u_1, u_2, \dots}{\text{minimize}} \lim_{K \rightarrow +\infty} \frac{\bar{K}}{K}, \quad (4)$$

where \bar{K} is the number of blocked requests, K is the total number slots, and the optimization variables are the SBS's actions $\{u_1, u_2, \dots\}$. We aim to find the optimal policy $\{\mu_1, \mu_2, \dots\}$, where the policy μ_k is a mapping from the state space to the action space, i.e., the action in slot k is $u_k = \mu_k(x_k)$, where the system state x_k will be detailed in the next section. According to the statistics of the system including the energy arrival distribution, content request generating rate, content popularity and updating process, we can find the optimal policy via dynamic programming, which will be detailed in the next section.

III. OPTIMAL POLICY WITH FINITE BATTERY CAPACITY

To find the optimal solution for the problem (4), we need to decide the SBS's action based on the system state at the beginning of each slot. MDP [30] is an effective mathematical framework to formulate this type of decision making problems. DP algorithm is widely used to deal with the control optimization of stochastic process by breaking it down into a collection of simpler per-stage¹ subproblems which only depends on the current system state. A standard MDP problem contains the following elements: states, actions, cost function, and state transition. Notice that

¹In this paper, the term "stage" is equivalent to the term "slot".

the energy states and the channel states are continuous in our problem. To make the problem tractable, we first discretize the system states, and then re-formulate our problem as an MDP optimization and use policy iteration algorithm to find the optimal policy.

A. System State

The system state in stage k is denoted by $x_k = (E_k, Q_k, I_k, C_k)$, where the energy state E_k is the amount of energy in the battery, the request state Q_k is the energy required for the unicast, the push state C_k is the number of pushed contents, and $I_k \in \{0, 1\}$ indicates which content is requested. $I_k = 1$ if the required content is c_{C_k+1} , and $I_k = 0$ if the rank of required content is larger than $C_k + 1$. The reason to introduce I_k is that if push action is taken, $(C_k + 1)$ -th ranked content c_{C_k+1} will be transmitted as we always push the most popular content in the list $\tilde{C}_k = (c_{C_k+1}, \dots, c_N)^T$. Thus the user request for c_{C_k+1} is simultaneously satisfied. Recall that Q_k represents the energy consumption for unicast, which is calculated as $Q_k = E_u(d) = P_t(d)T_p$, where $P_t(d)$ is the transmission power obtained by solving (1). And $Q_k = 0$ represents that there is no content request. Denote the state space as \mathcal{S} .

To make the DP algorithm tractable, we discretize the energy state E_k and the content request state Q_k into finite sets. Although, the optimal solution for the discretized finite state problem may not be optimal for the continuous state problem, the gap diminishes as the discretization becomes finer [30, Vol. I, Sec. 6.6.1]. Specifically, the energy is discretized with unit energy E_{unit} . Then the energy state can be denoted by $E_k \in \{0, 1, \dots, E_{\text{max}}\}$ with $E_{\text{max}}E_{\text{unit}} = B_{\text{max}}$. $E_k = i$ corresponds to iE_{unit} amount of energy in the battery, and similarly, $A_k = i$ corresponds to iE_{unit} amount of energy arrived in stage k .

To discretize Q_k , we select a series of distances $0 < d_1 < d_2 < \dots < d_M = R$ so that $P_t(d_i)T_p = l_i E_{\text{unit}}$ where l_i is a positive integer for any $i = 1, 2, \dots, M$. If a user's distance d to the SBS satisfies $d_{i-1} < d \leq d_i$, we unicast the requested content with energy $P_t(d_i)T_p$, which guarantees the average data rate r_0 for all the users in this area. And we set $l_0 = 0$ denoting that the required energy for unicast is zero. Then the content request state can be denoted by $Q_k \in \{0, 1, \dots, M\}$, where $Q_k = i$ corresponds to the case that $l_i E_{\text{unit}}$ amount of energy is required for unicast. With the discretization, the state space \mathcal{S} is of dimension $(E_{\text{max}} + 1) \times (2M + 1) \times (N + 1)$.

B. MDP Problem Formulation and Optimization

Besides the system state, we need to further clarify the action, the cost function and the state transition to complete the MDP problem formulation. The action has been modeled as (2). Notice that in different states, the SBS may not be able to take all the three actions. A simple example is that if $E_k = 0$, the SBS can do nothing but sleep, i.e., $u_k = 0$. Hence, the action space is state-dependent, which can be expressed as $u_k \in \mathcal{U}_k(x_k)$. If $E_k \geq l_{Q_k} \geq 1$, i.e., the energy for unicast is available, we have $1 \in \mathcal{U}_k(x_k)$. On the other hand, if $E_k \geq E_p$ and $C_k < N$, i.e., the energy for push is available and there are contents to be pushed, we have $2 \in \mathcal{U}_k(x_k)$.

The per-stage cost function $g_k(x_k, u_k)$ denotes whether a content request is blocked or not. Mathematically, it can be expressed as

$$g_k(x_k, u_k) = \begin{cases} 1, & \text{if } Q_k > 0, u_k \neq 1, \text{ and } I_k u_k = 0 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

Notice that a content request is blocked if there is a request but the requested content is not transmitted by either unicast or push. When $g_k(x_k, u_k) = 0$, it refers to either there is no unicast request or the requested content is in the pushed list C_k .

The state transition is expressed as the conditional probability

$$\begin{aligned} p_{x_k \rightarrow x_{k+1} | u_k} &= \Pr(E_{k+1}, Q_{k+1}, I_{k+1}, C_{k+1} | E_k, Q_k, I_k, C_k, u_k) \\ &= \Pr(E_{k+1} | E_k, Q_k, u_k) \Pr(C_{k+1} | C_k, u_k) \Pr(Q_{k+1}, I_{k+1} | C_{k+1}), \end{aligned} \quad (6)$$

where the second equality is derived by the definition of conditional probability. We calculate the state transition probability according to (6). Firstly, we denote $p_a(i), i = 0, 1, \dots$ as the probability that iE_{unit} amount of energy arrives, which satisfies $p_a(i) \in [0, 1]$ and $\sum_i p_a(i) = 1$. To simplify the description, we set $p_a(i) = 0, \forall i = -1, -2, \dots$. Then we have

$$\Pr(E_{k+1} | E_k, Q_k, u_k) = \begin{cases} p_a(E_{k+1} - E_k), & \text{if } u_k = 0, E_{k+1} < E_{\max} \\ 1 - \sum_{i=0}^{E_{\max} - E_k - 1} p_a(i), & \text{if } u_k = 0, E_{k+1} = E_{\max} \\ p_a(E_{k+1} - E_k + l_{Q_k}), & \text{if } u_k = 1, E_{k+1} < E_{\max} \\ 1 - \sum_{i=0}^{E_{\max} - E_k + l_{Q_k} - 1} p_a(i), & \text{if } u_k = 1, E_{k+1} = E_{\max} \\ p_a(E_{k+1} - E_k + l_M), & \text{if } u_k = 2, E_{k+1} < E_{\max} \\ 1 - \sum_{i=0}^{E_{\max} - E_k + l_M - 1} p_a(i), & \text{if } u_k = 2, E_{k+1} = E_{\max} \end{cases} \quad (7)$$

Notice that when $E_{k+1} = E_{\max}$, the energy arrival may exceed the battery capacity. So the probability is calculated by summarizing all the possible energy arrival conditions.

Secondly, the push state C_k can only transit to its neighboring values $C_k + 1, C_k - 1$ or keep constant, i.e., $C_{k+1} \in \{\max\{0, C_k - 1\}, C_k, \min\{N, C_k + 1\}\}$. The transition probability of push state C_k is expressed as

$$\Pr(C_{k+1}|C_k, u_k) = \begin{cases} p_c \frac{C_k}{N}, & \text{if } u_k < 2, C_{k+1} = C_k - 1 \geq 0, \text{ or } u_k = 2, C_{k+1} = C_k < N \\ 1 - p_c \frac{C_k}{N}, & \text{if } u_k < 2, C_{k+1} = C_k, \text{ or } u_k = 2, C_{k+1} = C_k + 1 \leq N \\ 0. & \text{else} \end{cases} \quad (8)$$

Finally, the content request state transition is

$$\Pr(Q_{k+1}, I_{k+1}|C_{k+1}) = \begin{cases} (1 - p_u) + p_u \sum_{i=1}^{C_{k+1}} f_i, & \text{if } Q_{k+1} = 0 \\ p_u f_{C_{k+1}+1} \frac{d_m^2 - d_{m-1}^2}{R^2}, & \text{if } Q_{k+1} = m > 0, \text{ and } I_{k+1} = 1 \\ p_u (1 - \sum_{i=1}^{C_{k+1}+1} f_i) \frac{d_m^2 - d_{m-1}^2}{R^2}, & \text{if } Q_{k+1} = m > 0, \text{ and } I_{k+1} = 0 \end{cases} \quad (9)$$

where f_i is calculated according to (3) and $d_0 = 0$. The transition probability (9) is calculated based on our model, i.e., there is a content request with probability p_u , requesting which content depends on the popularity distribution f_i , and the location of the request follows uniform distribution in the cell coverage. Notice that if $C_{k+1} = N$, we always have $Q_{k+1} = 0$. In summary, the state transition probability (6) is calculated based on the widely-used models about energy arrival, content popularity, and user distribution. In fact, our framework is also applicable to some other models.

Based on the MDP framework, the original optimization problem (4) can be re-written as

$$\min \lim_{K \rightarrow +\infty} \frac{1}{K} \mathbb{E} \left[\sum_{k=0}^{K-1} g(x_k, \mu_k(x_k)) \right]. \quad (10)$$

The expectation operation is taken over all the random parameters. The optimization is taken over all the possible policies $\{\mu_1, \mu_2, \dots\}$. It can be proved that for any two states, there is a stationary policy $\mu = \{\mu(x)\}_{x \in \mathcal{S}}$ so that one state can be accessed with non-zero probability from the other with finite steps. Consequently, the optimal value is independent of the initial state x_0 , and there exists an optimal stationary policy μ^* [30, Vol. II, Sec 4.2].

According to [30, Vol. II, Prop. 4.2.1], the optimal average cost λ^* together with some vector $h^* = \{h^*(x)|x \in \mathcal{S}\}$ satisfies the Bellman's equation

$$\lambda^* + h^*(x) = \min_{u \in \mathcal{U}(x)} \left[g(x, u) + \sum_{y \in \mathcal{S}} p_{x \rightarrow y|u} h^*(y) \right]. \quad (11)$$

Further more, if $u = \mu^*(x)$ attains the minimum value of (11) for each x , the stationary policy μ^* is optimal. Based on the Bellman's equation, instead of the long term average cost minimization, we only need to deal with (11) which only relates with per-stage cost $g(x, u)$ and state transition probability $p_{x \rightarrow y|u}$. The policy iteration algorithm [30, Vol. II, Sec. 4.4] can effectively solve the problem.

C. Policy Iteration Algorithm

The policy iteration algorithm starts with any feasible stationary policy, and improves the objective step by step. Suppose in the j -th step, we have a stationary policy denoted by $\mu^{(j)}$. Based on this policy, we perform *policy evaluation* [30, Vol. II, Sec. 4.4], i.e., we solve the following linear equations

$$\lambda^{(j)} + h^{(j)}(x) = g(x, \mu^{(j)}(x)) + \sum_{y \in \mathcal{S}} p_{x \rightarrow y|\mu^{(j)}(x)} h^{(j)}(y) \quad (12)$$

for $\forall x \in \mathcal{S}$ to get the average cost $\lambda^{(j)}$ and vector $h^{(j)}$. Notice that there are $(E_{\max} + 1) \times (2M + 1) \times (N + 1)$ equations but $(E_{\max} + 1) \times (2M + 1) \times (N + 1) + 1$ unknown parameters, hence more than one solutions exist, which are different with each other by a constant value for all $h^{(j)}(x)$. Without loss of generality, we can set for example $h^{(j)}(E_{\max} + 1, 2M + 1, N_S + 1) = 0$, then the solution for (12) is unique.

As $\mu^{(j)}$ may not be the optimal policy, we subsequently perform *policy improvement* [30, Vol. II, Sec. 4.4] step to find the policy $\mu^{(j+1)}$ which minimizes the right hand side of Bellman's equation

$$\mu^{(j+1)}(x) = \arg \min_{u \in \mathcal{U}(x)} \left[g(x, u) + \sum_{y \in \mathcal{S}} p_{x \rightarrow y|u} h^{(j)}(y) \right]. \quad (13)$$

If $\mu^{(j+1)} = \mu^{(j)}$, the algorithm terminates, and the optimal policy is obtained $\mu^* = \mu^{(j)}$. Otherwise, repeat the procedure by replacing $\mu^{(j)}$ with $\mu^{(j+1)}$. It is proved that the policy iteration algorithm terminates in finite number of iterations [30, Vol. II, Prop. 4.4.1].

Although the DP-based solution can obtain the optimal policy, it has two drawbacks. Firstly, it usually encounters the *curse of dimensionality* [30] when the number of states is quite large. Secondly, it is difficult to obtain the closed-form expressions to reveal the structure of the optimal policy. The algorithm need to be operated for every set of parameters to get the numerical result. To tackle these problems, we try to design some heuristic algorithms in the next section. Notice

that still, studying MDP formulation is meaningful as its optimality provides a theoretical upper-bound to evaluate the performance of the other sub-optimal algorithms. In addition, the optimal policy can be efficiently solved by the DP algorithm if the number of system states is relatively small.

IV. THRESHOLD-BASED POLICIES

In this section, we consider some threshold-based policies. The intuition is that pushing the popular contents enhances the energy efficiency as lots of duplicated unicasts can be avoided. However, the push action itself consumes energy. As we always push the most popular contents, there must be a threshold on C_k , denoted by C_{thr} , so that the energy consumption for pushing contents in the list $\mathbf{C} = (c_1, \dots, c_{C_{\text{thr}}})$ is less than the total energy consumption for duplicated unicasts without push. Also, threshold-based policies are analytically tractable and have some good asymptotic properties, as shown later. Thus, we consider that the SBS always takes push action as long as the number of pushed contents is smaller than the threshold C_{thr} . Notice that as the system is energy limited, the threshold should depend on the energy state E_k as well as the content request state Q_k . However, it is difficult to analyze the state-dependent thresholds. Hence, we focus on the policies with constant threshold to obtain some closed-form results.

There are two steps to design a threshold-based policy. Firstly, determine the threshold C_{thr} . Secondly, determine the action that is taken when the threshold C_{thr} is achieved. With the objective of minimizing the number of requests blocked by the SBS, we first analyze the performance of push action with sufficient energy, which is detailed in the following lemmas. Notice that our analysis in this section is done in the continuous regime.

Lemma 1. *If the required energy for push can always be satisfied, the stationary probability with which the SBS pushes a content in threshold-based policy is*

$$\Pr(u_k = 2) = \frac{p_c C_{\text{thr}}}{N + p_c}. \quad (14)$$

Proof: See Appendix A. ■

The blocking probability of a threshold-based policy is composed of two parts. The first part is due to push action itself. If the SBS is pushing a content while a user is requesting another content, the request is blocked. The second part is due to energy shortage, i.e., the user request

will be blocked if there is not enough energy for unicast. Based on the observation, we can derive the performance lower bound.

Lemma 2. *The blocking probability of the threshold-based policy is lower bounded by*

$$p_{\text{blk, LB}}(C_{\text{thr}}) = \frac{p_c p_u C_{\text{thr}}}{N + p_c} \left(1 - \sum_{i=1}^{C_{\text{thr}}} f_i\right). \quad (15)$$

Proof: See Appendix B. ■

The lower bound (15) is achievable when the required energy for unicast can always be satisfied. While on the other hand, there are some other policies to achieve zero blocking probability when the power supply is sufficient. For example, the SBS can always take unicast action. In this case, the threshold-based policies may not be optimal. However, with insufficient power supply, threshold-based policies can still provide substantial benefits. We will show the performance later by simulations. In the rest of this section, we will analyze several threshold-based policies proposed from different points of view.

A. Push-Only Threshold-based Policy

Notice that the more the contents are pushed to the user side, the fewer the unicast requests are triggered. Consequently, the number of blocking events due to energy shortage is reduced. Based on this, we consider the POTB policy in which all the harvested energy is used to push contents. If the threshold is achieved, the SBS does not take unicast action but turns to sleep to store energy. According to Lemma 1, the threshold for the POTB policy can be chosen according to

$$\frac{p_c C_{\text{thr, PO}}}{N + p_c} E_p \leq \bar{A}, \quad (16)$$

which means that the average energy consumption for push does not exceed the available harvested energy. Hence, we have

$$C_{\text{thr, PO}} = \min \left\{ N, \left\lfloor \frac{(N + p_c) \bar{A}}{p_c E_p} \right\rfloor \right\}, \quad (17)$$

where $\lfloor x \rfloor$ is the largest integer smaller than x . The performance of the POTB policy can be guaranteed by the following theorem.

Theorem 1. *Given the average energy arrival rate \bar{A} , if $B_{\max} \rightarrow +\infty$, the blocking probability of POTB policy is*

$$p_{\text{blk},\text{PO}} = p_u \left(1 - \sum_{i=1}^{C_{\text{thr},\text{PO}}} f_i \right), \quad (18)$$

where $C_{\text{thr},\text{PO}}$ is expressed as (17).

Proof: See Appendix C. ■

Notice that (18) is equivalent to the blocking probability with average power constraint. It can be considered as an extension of the result in [36] where the AWGN channel capacity with infinite battery is equal to that with average power constraint. When \bar{A} is strictly larger than the required energy to push C_{thr} contents but is not enough to push $C_{\text{thr}} + 1$ contents, the remained energy can be used for unicast. In this way, the performance can be slightly improved. Based on Theorem 1, if \bar{A} is large enough so that $C_{\text{thr},\text{PO}} = N$, we have $\sum_{i=1}^{C_{\text{thr},\text{PO}}} f_i = 1$ and hence, the blocking probability is 0. Otherwise, $p_{\text{blk},\text{PO}} > p_{\text{blk},\text{LB}}$ holds.

As (18) is a decreasing function of $C_{\text{thr},\text{PO}}$, the performance is better with larger threshold. It motivates us to design an improved version of POTB policy, named as *always-push threshold-based (APTB) policy*, which is equivalent to the POTB policy with threshold $C_{\text{thr}} = N$. However, if $\bar{A} < \frac{p_c N}{N + p_c} E_p$, there is not closed-form blocking probability expression for APTB policy as it depends on the distribution of energy arrival process.

B. Energy-Efficient Threshold-based Policy

There is a tradeoff between the energy consumed for content push and the energy saved by the reduction of duplicated unicasts. From energy-efficiency point of view, the push action should be taken if the energy saved exceeds the energy consumed. The following proposition describes the condition in which content push is the most energy-efficient.

Lemma 3. *If the index of content c_i satisfies*

$$i \leq \left(\frac{N p_u \bar{E}_u}{p_c E_p \sum_{j=1}^N 1/j^v} \right)^{\frac{1}{v}}, \quad (19)$$

where $\bar{E}_u = \int_0^R E_u(d) \frac{2d}{R^2} dd$ is the expected energy consumption of unicast, the energy consumption to push the content c_i is no more than the expected energy consumption for unicasting c_i upon requests.

Proof: See Appendix D. ■

Lemma 3 tells us that if the popularity rank of a content satisfies (19), it is more energy efficient to push it to the user side in advance than to unicast it when required. Denote the threshold by

$$C_{\text{thr,EE}} = \min \left\{ N, \left\lfloor \left(\frac{N p_u \bar{E}_u}{p_c E_p \sum_{j=1}^N 1/j^v} \right)^{\frac{1}{v}} \right\rfloor \right\}. \quad (20)$$

Then pushing all the contents with rank $i \leq C_{\text{thr,EE}}$ is optimal from the energy-efficiency perspective.

With limited power supply from the renewable energy, it is natural to use the harvested energy in an efficient way. For the push action, it is more energy efficient to just push the contents with popularity rank $i \leq C_{\text{thr,EE}}$. And for the unicast action, sending a content to a user closer to the SBS is more energy efficient. Based on the above intuition, the EETB policy works as follows. The SBS firstly guarantees that the contents $c_1, \dots, c_{C_{\text{thr,EE}}}$ are pushed to the user side, and then responds part of the unicast requests for the contents $c_{C_{\text{thr,EE}}+1}, \dots, c_N$ according to the users' location. It is detailed in Algorithm 1. In this algorithm, (23) is derived from the condition that the average energy consumption cannot exceed the average energy arrival, i.e.,

$$\frac{p_c C_{\text{thr,EE}} E_p}{N + p_c} + \eta \int_0^{\bar{d}} E_u(d) \frac{2d}{R^2} dd \leq \bar{A}, \quad (21)$$

where

$$\eta = \left(1 - \frac{p_c C_{\text{thr,EE}}}{N + p_c} \right) p_u \left(1 - \sum_{i=1}^{C_{\text{thr,EE}}} f_i \right) \quad (22)$$

can be viewed as the probability of generating a unicast request for a content c_i , where $i \in \{C_{\text{thr,EE}} + 1, \dots, N\}$, and the max and min operations in (23) are used so that if $\bar{A} \leq \frac{p_c C_{\text{thr,EE}} E_p}{N + p_c}$ or $\bar{A} \geq \frac{p_c C_{\text{thr,EE}} E_p}{N + p_c} + \eta \bar{E}_u$, the equation still works. Also notice that we adopt the distance metric to decide whether to unicast or not, which can be replaced by the measured average SINR in real system.

The blocking probability performance of the EETB policy is summarized as follows.

Theorem 2. If $\bar{A} \geq \frac{p_c C_{\text{thr,EE}} E_p}{N + p_c}$ and $B_{\text{max}} \rightarrow +\infty$, the blocking probability of EETB policy is

$$p_{\text{blk,EE}} = p_{\text{blk,LB}}(C_{\text{thr,EE}}) + \eta \frac{R^2 - \tilde{d}^2}{R^2}, \quad (24)$$

where $C_{\text{thr,EE}}$ is expressed as (20).

Algorithm 1 EETB policy

Find the maximum \tilde{d} satisfying

$$\int_0^{\tilde{d}} E_u(d) \frac{2d}{R^2} dd \leq \min\{\bar{E}_u, \frac{1}{\eta} \max\{0, \bar{A} - \frac{p_c C_{\text{thr,EE}} E_p}{N + p_c}\}\} \quad (23)$$

if $C_k < C_{\text{thr,EE}}$ **then**

if $E_k \geq E_p$ **then** Push a content to all the users.

else Keep sleep to store energy.

end if

else

if $0 < Q_k \leq \min\{E_u(\tilde{d}), E_k\}$ **then** Unicast the requested content to the user.

else Keep sleep to store energy.

end if

end if

Proof: See Appendix E. ■

The result is similar to Theorem 1, i.e., the blocking probability with infinite battery is equal to that with average power constraint. In this policy, the lower bound $p_{\text{blk,LB}}$ can be achieved if $\tilde{d} = R$, i.e., all the unicast requests are satisfied. Notice that if the condition $\bar{A} \geq \frac{p_c C_{\text{thr,EE}} E_p}{N + p_c}$ is not satisfied, the EETB policy degenerates to the APTB policy.

With the closed-form expressions (18) and (24), we can easily find a better policy to minimize the blocking probability, i.e, we adopt the corresponding policy with smaller value calculated based on (18) and (24).

C. Greedy Optimal Threshold-based Policy

In the above discussion, the thresholds of the POTB policy and the EETB policy are found by minimizing the number of unicast requests and by maximizing the energy efficiency, respectively. We can also find the optimal threshold-based policy by greedily searching over all the possible thresholds. For each threshold, we calculate the average energy required to maintain the threshold. Secondly, we use the remained energy for energy-efficient unicast, i.e., determine a radius d_{thr} and serve the content requests generated inside the circle. The procedure is detailed in Algorithm 2.

Algorithm 2 GOTB policy

```

1: Set  $C_{\text{thr}} = 0, p_{\text{blk}} = 1, d_{\text{thr}} = 0$ .
2: for  $C = 0$  to  $C_{\text{thr,PO}}$  do
3:   With threshold  $C$ , calculate  $\tilde{d}$  by (23), and then calculate  $p_{\text{blk},C}$  by (24).
4:   if  $p_{\text{blk},C} < p_{\text{blk}}$  then
5:     Set  $C_{\text{thr}} = C, p_{\text{blk}} = p_{\text{blk},C}, d_{\text{thr}} = \tilde{d}$ .
6:   end if
7: end for

```

When the algorithm terminates, the threshold-based policy with parameters $C_{\text{thr}}, d_{\text{thr}}$ is optimal. Notice that the search range is from 0 to $C_{\text{thr,PO}}$ since when $C > C_{\text{thr,PO}}$, there is no spare energy for unicast. The GOTB policy can be used to evaluate the performance of the POTB policy and the EETB policy. Also notice that all the proposed threshold-based policies only depend on the statistic information of energy, contents, and user traffic. Hence, they are applicable to some other general models.

D. Analysis with Finite Battery Capacity

With finite battery capacity, the closed-form blocking probability expression cannot be obtained. We analyze the threshold-based policies using FSMC model. Specifically, for a given stationary policy $\mu(x)$, the state transition $p_{x_k \rightarrow x_{k+1} | \mu(x_k)}$ can be calculated based on (6) and the equations thereafter. Then we can get the state transition matrix \mathbf{P}_μ . By solving $\pi \mathbf{P}_\mu = \pi$ and $\sum_x \pi_x = 1, \pi_x \geq 0$, the stationary distribution of the system states can be obtained. Finally, the blocking probability can be calculated as $p_{\text{blk}} = \sum_{x \in \{x: Q > 0, u \neq 1 \text{ and } I_u = 0\}} \pi_x$.

Notice that for the EETB policy, \tilde{d} in (23) cannot be arbitrary real value due to the discretization on the content request state. In stead, we have $\tilde{d} \in \{0, d_1, \dots, d_M\}$. As a result, there might be a gap between the available energy and the energy required for unicast, which causes some quantization error.

V. NUMERICAL RESULTS

We run some simulations to evaluate the performance of proposed algorithms. We set the cell radius $R = 50\text{m}$, the required content delivery spectrum efficiency $r_0/W = 1\text{bps/Hz}$, the pathloss

parameters $\beta = 10\text{dB}$ and $\alpha = 2$. The maximum transmit power or equivalently the transmit power for cell-edge user is set $P_t(R) = 1\text{Watt}$. The channel coefficient h follows Rayleigh fading, whose mean value and the interference plus noise power $\sigma^2 + I$ are set so that (1) holds for $r = r_0, d = R, P_t = P_t(R)$. The users are sorted into $M = 5$ classes and the energy is quantized with unit $E_{\text{unit}} = \frac{P_t(R)T_p}{M}$. Hence, the number of energy units for push is $E_p = M$, and the content request in class m consumes m units of energy. Assume the energy arrival process follows a Poisson distribution. Notice that our analytical results do not depend on the energy arrival model, and hence can be applied to some more realistic models. We set $N = 20$, and the Zipf parameter $v = 1$.

Firstly, we set $\bar{A} = 1.0, p_u = 0.9$ in the simulations and evaluate the influence of the battery capacity. The results are depicted in Figs. 3 and 4. Comparing the two figures, it can be found that the POTB policy is more tolerable to the limited battery capacity than the EETB policy. On the one hand, the performance degradation of the POTB policy w.r.t. the infinite battery case is smaller than that of the EETB policy. For instance, with $p_c = 0.4$ and $E_{\text{max}} = 20$, the blocking probability of the POTB policy increases by 10%, while the EETB policy increases by more than 40%. On the other hand, the POTB policy converges to the infinite battery case much faster than the EETB policy. Take $p_c = 0.6$ as an example, the convergence point of the POTB policy is $E_{\text{max}} \approx 50$, but that of the EETB policy is larger than 500. The reason is that if we view the battery as an energy queue, the service process of the POTB policy (only push) is less dynamic than that of the EETB policy (including both push and unicast with variable energy requirements).

Then we evaluate the performance of the POTB policy, the EETB policy as well as the APTB policy by comparing with the GOTB policy, the DP optimal policy and the no-push baseline policy which is termed as *service-on-demand* policy, i.e., contents are not pushed and the users are served upon requests.

Fig. 5 shows the performance comparison by varying the content updating rate p_c . We set the energy arrival rate $\bar{A} = 1.5$ and the content request generating rate $p_u = 0.9$. Compared with the service-on-demand policy, the push policies perform better and the gain is enhanced as the content updating rate p_c decreases, which indicates that the benefit of push is more noticeable if the content set is more stable. On the other hand, there is a cross point between the EETB policy and the POTB policy. The EETB policy achieves the same performance with the GOTB

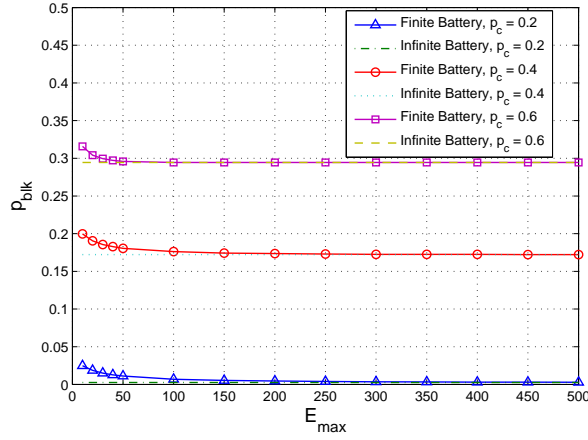


Fig. 3. The influence of battery capacity on the POTB policy. $\bar{A} = 1.0$, $p_u = 0.9$.

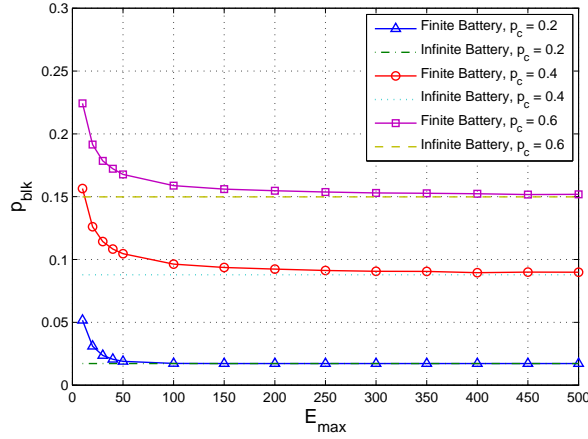


Fig. 4. The influence of battery capacity on the EETB policy. $\bar{A} = 1.0$, $p_u = 0.9$.

policy when $p_c \geq 0.4$, while the POTB policy performs the same when $p_c \leq 0.3$. While they converge to the same when $p_c \leq 0.1$ since $C_{\text{thr},\text{PO}} = C_{\text{thr},\text{EE}} = N$ in this situation. Also, the APTB policy shows a slight performance gain compared with the POTB policy. However, there is still gap between the threshold-based policies and the DP optimal policy.

We further compare the policies by varying the energy arrival rate $\bar{A} = A$ with $p_c = 0.2$ and $p_u = 0.9$. The result is illustrated in Fig. 6. As the content updating rate is relatively small, the performance gain of push-based policies is remarkable compared with the non-push policy. The APTB policy performs the same with the GOTB policy, while the POTB policy

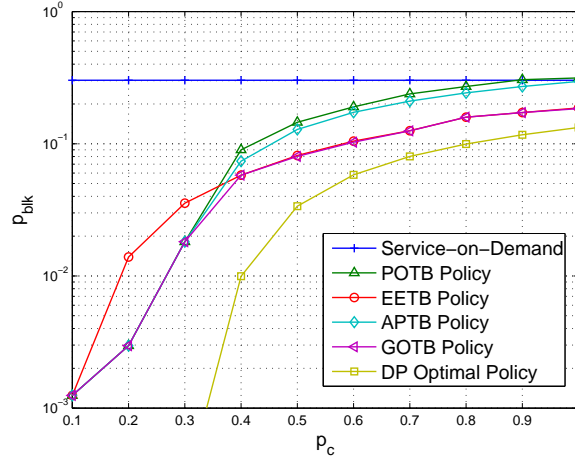


Fig. 5. The performance comparison versus the content updating rate p_c with $N = 20$. The energy arrival rate is $\bar{A} = 1.5$ and the content request generating rate is $p_u = 0.9$.

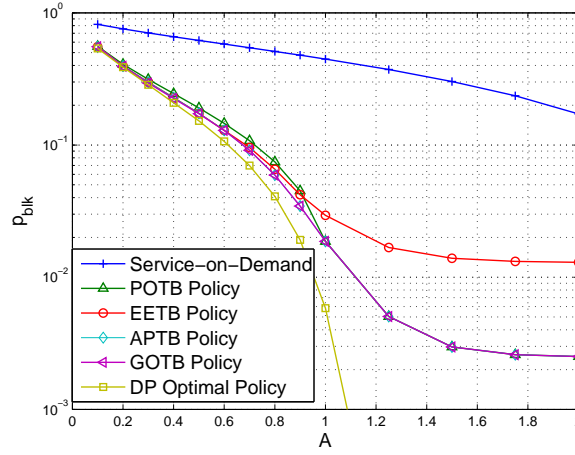


Fig. 6. The performance comparison versus the energy arrival rate $\bar{A} = A$. The content updating rate is $p_c = 0.2$, and the content request generating rate is $p_u = 0.9$.

and the EETB policy perform the same as the GOTB policy in high energy arrival rate regime and low energy arrival rate regime, respectively. Notice that the curves corresponding to the threshold-based policies become flat as \bar{A} is large enough, which illustrates the performance lower bound. Finally, compared with the DP optimal policy, the performance gap between it and the threshold-based policies diminishes when \bar{A} becomes small.

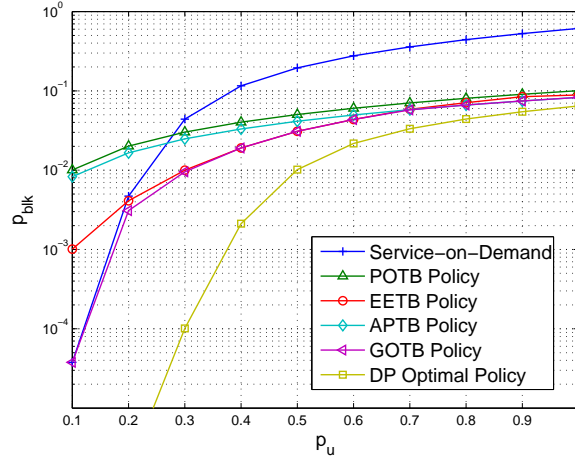


Fig. 7. The performance comparison versus the content request generating rate p_u . The energy arrival rate is $\bar{A} = 0.75$, and the content updating rate is $p_c = 0.2$.

At last, the performance variation w.r.t. the content request generating rate p_u is depicted in Fig. 7. Notice that except the GOTB policy, our proposed threshold-based policies perform worse than the service-on-demand policy when the content request generating rate is very low. This is because if the content request arrives very slowly (even slower than the content updating rate), each content is requested very few times (no larger than 1 with high probability). It is meaningless to push contents in advance. However, as the search process in the GOTB policy contains the case $C = 0$, it will ultimately converge to the service-on-demand policy as p_u decreases. Compared with the GOTB policy, there is a cross point between the EETB policy curve and the APTB policy curve. The former performs the same with the GOTB policy with low content request generating rate (lower than 0.7 but not lower than 0.3), the latter does with high content request generating rate (higher than 0.7). In addition the performance gap between the threshold-based policies and the DP optimal policy becomes small as p_u increases.

VI. CONCLUSION

In this paper, proactive push policies in EH based SBS were studied. With infinite battery capacity, two threshold-based policies, the POTB policy and the EETB policy, achieve the closed-form blocking probability the same with average power constraint. With finite battery capacity, we compared the threshold-based policies with the optimal policy and the conventional service-

on-demand policy. From the numerical simulations, we found that tremendous performance gain can be obtained through push when the content updating rate is low or the content request generating rate is high. We also find that there are cross points between the POTB policy and the EETB policy. The POTB policy performs better than the EETB policy if the energy arrival rate is high or the content updating rate is low. In addition, the POTB policy is more tolerable to the limited battery capacity than the EETB policy. Besides, the threshold-based policies perform close to the optimal policy in the following cases: the energy arrival rate is low, or the content request generating rate is high. Future work can consider communication overheads of push mechanism, such as the content fetching and caching cost at the SBS, the content storage cost at the users, and etc. Besides, inter-SBS cooperative content caching and push in multiple SBSs scenario will also be an interesting future research direction.

APPENDIX A

PROOF OF LEMMA 1

If the energy for push is always satisfied, C_k will not be smaller than $C_{\text{thr}} - 1$, since every time that $C_k = C_{\text{thr}} - 1$, a new content is pushed ($u_k = 2$). Thus, there are in total two stationary push states $C_k \in \{C_{\text{thr}}, C_{\text{thr}} - 1\}$, and the push action is always taken in state $C_k = C_{\text{thr}} - 1$. We have the transition probabilities $\Pr(C_{k+1} = C_{\text{thr}} - 1 | C_k = C_{\text{thr}}) = p_c \frac{C_{\text{thr}}}{N}$, $\Pr(C_{k+1} = C_{\text{thr}} - 1 | C_k = C_{\text{thr}} - 1) = p_c \frac{C_{\text{thr}} - 1}{N}$. As a result, the push state transition matrix can be expressed as

$$\mathbf{P}_C = \begin{pmatrix} p_c \frac{C_{\text{thr}} - 1}{N} & 1 - p_c \frac{C_{\text{thr}} - 1}{N} \\ p_c \frac{C_{\text{thr}}}{N} & 1 - p_c \frac{C_{\text{thr}}}{N} \end{pmatrix}. \quad (25)$$

By solving the equilibrium equation $\boldsymbol{\pi} = \boldsymbol{\pi} \mathbf{P}_C$, where $\boldsymbol{\pi} = (\pi_{C_{\text{thr}} - 1}, \pi_{C_{\text{thr}}})$, $\pi_i = \Pr(C_k = i)$, $i = C_{\text{thr}} - 1, C_{\text{thr}}$ are the stationary probabilities of push state, we can get

$$\Pr(u_k = 2) = \pi_{C_{\text{thr}} - 1} = \frac{p_c C_{\text{thr}}}{N + p_c}. \quad (26)$$

APPENDIX B

PROOF OF LEMMA 2

We derive the lower bound by assuming that there is sufficient power supply for any actions. Thus, a request is blocked only when the SBS is pushing a content while simultaneously another

content is requested. Depending on (9) and (14), the stationary blocking probability satisfies

$$\begin{aligned}
 p_{\text{blk}} &\geq \Pr(Q_k > 0, I_k = 0, u_k = 2) \\
 &= \Pr(u_k = 2)\Pr(Q_k > 0, I_k = 0|u_k = 2) \\
 &= \frac{p_c C_{\text{thr}}}{N + p_c} p_u \left(1 - \sum_{i=1}^{C_{\text{thr}}} f_i\right) \stackrel{\text{def}}{=} p_{\text{blk, LB}}(C_{\text{thr}}).
 \end{aligned} \tag{27}$$

APPENDIX C

PROOF OF THEOREM 1

For simplicity, we denote $C = C_{\text{thr, PO}}$. Firstly, we assume that there is sufficient power supply for push. In this case, the request for a content in the remained list $\tilde{C}_k = (c_{C+1}, \dots, c_N)$ will be blocked. Obviously, the stationary blocking probability is expressed as (18).

Next, we prove that the blocking probability with energy harvesting and infinite battery capacity is also (18). It is equivalent to show that the stationary energy shortage probability, denoted by p_s , is zero. We prove it by contradiction. Specifically, we assume that

$$p_s = \Pr(E_k < E_p | C_k = C - 1) > 0 \tag{28}$$

and find the contradiction.

We consider the content updating process and the energy process separately. For the content updating process, we introduce the concept of *push request indicator* denoted by D_k . When a pushed content leaves, a new content needs to be pushed, i.e., a push request is generated. We use $D_k = 1$ to indicate that a push request is generated, and $D_k = 0$ otherwise. The push state achieves C only if all the push requests are satisfied. By definition, when a push request is generated, it is equivalent to that a pushed content leaves. Therefore, a push request will be generated with probability $\Pr(D_{k+1} = 1 | C_k = i) = \frac{p_c i}{N}, i \leq C$.

Denote the stationary distribution of C_k by $\pi_i = \Pr(C_k = i), i = 0, \dots, C$, which satisfies $\pi_i \geq 0$ and $\sum_{i=0}^C \pi_i = 1$. By the law of total probability, the stationary probability of push request can be calculated as

$$\begin{aligned}
 \Pr(D_k = 1) &= \sum_{i=0}^C \Pr(C_{k-1} = i) \Pr(D_k = 1 | C_{k-1} = i) = \sum_{i=0}^C \pi_i \frac{p_c i}{N} \\
 &\leq \pi_C \frac{p_c C}{N} + \sum_{i=0}^{C-1} \pi_i \frac{p_c (C-1)}{N} = \pi_C \frac{p_c}{N} + \frac{p_c (C-1)}{N},
 \end{aligned} \tag{29}$$

where π_C can be calculated by solving the equilibrium equation $\pi = \pi \mathbf{P}_C$, where $\pi = (\pi_0, \dots, \pi_C)$, and the elements (denoted by $p_{i,j} = \Pr(C_{k+1} = j | C_k = i)$, $0 \leq i \leq C, 0 \leq j \leq C$) of the state transition probability matrix \mathbf{P}_C satisfy that: (1) $\sum_j p_{i,j} = 1, p_{i,j} \geq 0, \forall i$; (2) if $|i - j| > 1$, $p_{i,j} = 0$. The expression of π_C is

$$\pi_C = \left(1 + \frac{p_{C,C-1}}{p_{C-1,C}} + \dots + \frac{\prod_{n=1}^C p_{n,n-1}}{\prod_{n=1}^C p_{n-1,n}} \right)^{-1}. \quad (30)$$

We compare with sufficient energy input case as in Appendix A, in which only two stationary states $C_k \in \{C, C-1\}$ are accessible, and the push request generating probability can be calculated similarly

$$\Pr'(D_k = 1) = \pi'_C \frac{p_c}{N} + \frac{p_c(C-1)}{N}, \quad (31)$$

where the stationary distribution

$$\pi'_C = \left(1 + \frac{p'_{C,C-1}}{p'_{C-1,C}} \right)^{-1} \quad (32)$$

is obtained by solving $\pi' = \pi' \mathbf{P}_C'$ where the elements of transition probability matrix \mathbf{P}_C' satisfies $p'_{i,j} = 0, \forall i < C-1, j < C-1$. As

$$p'_{C,C-1} = p_{C,C-1} = p_c \frac{C}{N}, \quad (33)$$

based on our assumption $p_s > 0$, we have

$$\begin{aligned} p_{C-1,C} &= p_s \Pr(C_{k+1} = C | C_k = C-1, E_k < E_p) + (1-p_s) \Pr(C_{k+1} = C | C_k = C-1, E_k > E_p) \\ &= p_s \cdot 0 + (1-p_s) p'_{C-1,C} < p'_{C-1,C}. \end{aligned} \quad (34)$$

Comparing (30) and (32), we have $\pi'_C > \pi_C$, which results in $\Pr(D_k = 1) < \Pr'(D_k = 1)$ according to (29) and (31). On the other hand, we have $\pi'_C = 1 - \frac{p_c C}{N+p_c}$ as derived in Appendix A. Thus, we have

$$\Pr(D_k = 1) < \frac{p_c C}{N + p_c}. \quad (35)$$

For energy process, each push request needs to be satisfied with energy usage of E_p . The average energy consumption for push satisfies

$$\Pr(D_k = 1) E_p < \frac{p_c C}{N + p_c} E_p \leq \bar{A}. \quad (36)$$

If we consider the energy buffer as a queuing system where the energy units are viewed as customers, (36) tells that the customer (energy) arrival rate \bar{A} is strictly larger than the service (energy consumption) rate. In this case, the queuing system is unstable, i.e., the arrived customers cannot all be served and the queue overflows [37, pp. 4]. As a result, the energy shortage probability is zero, i.e., $p_s = 0$, which contradicts with the assumption (28). Hence, the theorem is proved.

APPENDIX D

PROOF OF LEMMA 3

Denote N_p as the number of slots in which a content stays in the content list \mathcal{C} . According to the content updating process, N_p follows the geometric distribution, i.e.,

$$\Pr(N_p = n) = (1 - p_c \frac{1}{N})^{n-1} p_c \frac{1}{N}. \quad (37)$$

Denote N_r as the number of requests for the content c_i . In the case that the content c_i stays in the system for n slots, N_r follows the binomial distribution

$$\Pr(N_r = k | N_p = n) = \binom{n}{k} (p_u f_i)^k (1 - p_u f_i)^{n-k}. \quad (38)$$

By the law of total expectation, the average number of requests for the content with rank i is

$$\begin{aligned} \mathbb{E}[N_r] &= \mathbb{E}_{N_p}[\mathbb{E}[N_r | N_p]] \\ &= \sum_{n \geq 1} \Pr(N_p = n) n p_u f_i \\ &= \mathbb{E}[N_p] p_u f_i = \frac{N}{p_c} p_u f_i. \end{aligned} \quad (39)$$

Since the channel state is independent of the content request, the expected unicast energy consumption for all these requests is

$$\begin{aligned} \mathbb{E}[E_u] &= \mathbb{E}[N_r] \bar{E}_u \\ &= \frac{N}{p_c} p_u f_i \bar{E}_u. \end{aligned} \quad (40)$$

If the content of rank i is pushed to the users, the requests for this content are satisfied by the users' local storage. Hence, the unicast energy consumption (40) is avoided. It can be found that pushing the content of rank i can reduce the overall energy consumption if

$$E_p \leq \frac{N}{p_c} p_u f_i \bar{E}_u. \quad (41)$$

With some simple derivation, the condition (19) is obtained.

APPENDIX E

PROOF OF THEOREM 2

We prove the theorem by contradiction similar to Appendix C. Denote $C = C_{\text{thr,EE}}$. If the required energy for EETB policy is always available, a request is blocked when either the SBS is pushing a content, or the user's distance to the SBS is larger than \tilde{d} . Thus the stationary blocking probability is

$$p_{\text{blk}} = \Pr(Q_k > 0, I_k = 0, u_k = 2) + \Pr(u_k \neq 2)\Pr(Q_k > E_u(\tilde{d})|u_k \neq 2), \quad (42)$$

which results in (24). Then we assume the non-zero energy shortage probability and find the contradiction. The difference with Appendix C is that the energy shortage may happen in either push state or unicast state. Firstly, we assume the energy is insufficient for push, i.e., $p_s = \Pr(E_k < E_p|C_k = C - 1) > 0$. Similar with Appendix C, we can infer that $\pi_C < \pi'_C$ due to $p_s > 0$, and the required energy satisfies

$$\begin{aligned} & \Pr(D_k = 1)E_p + \pi_C p_u \left(1 - \sum_{i=1}^C f_i\right) \int_0^{\tilde{d}} E_u(d) \frac{2d}{R^2} dd \\ & < \frac{p_c C}{N + p_c} E_p + \pi'_C p_u \left(1 - \sum_{i=1}^C f_i\right) \int_0^{\tilde{d}} E_u(d) \frac{2d}{R^2} dd \leq \bar{A}. \end{aligned} \quad (43)$$

Secondly, we assume the energy is insufficient for unicast, i.e., $\int_0^{\tilde{d}} p_d(d) dd > 0$, where $p_d(d) = \Pr(E_k < E_u(d)|Q_k = E_u(d), d > 0)$. Then the required energy satisfies

$$\frac{p_c C}{N + p_c} E_p + \eta \int_0^{\tilde{d}} (1 - p_d(d)) E_u(d) \frac{2d}{R^2} dd < \bar{A}. \quad (44)$$

In both cases, the energy queue overflows. Hence, both (43) and (44) contradict the non-zero energy shortage probability assumption, and the theorem is proved.

REFERENCES

- [1] D. Gunduz, K. Stamatiou, N. Michelusi, and M. Zorzi, "Designing intelligent energy harvesting communication systems," *IEEE Communications Magazine*, vol. 52, no. 1, pp. 210–216, Jan. 2014.
- [2] V. Sharma, U. Mukherji, V. Joseph, and S. Gupta, "Optimal energy management policies for energy harvesting sensor nodes," *IEEE Transactions on Wireless Communications*, vol. 9, no. 4, pp. 1326–1336, Apr. 2010.

- [3] J. Yang and S. Ulukus, "Optimal packet scheduling in an energy harvesting communication system," *IEEE Transactions on Communications*, vol. 60, no. 1, pp. 220–230, Jan. 2012.
- [4] O. Ozel, K. Tutuncuoglu, J. Yang, S. Ulukus, and A. Yener, "Transmission with energy harvesting nodes in fading wireless channels: Optimal policies," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 8, pp. 1732–1743, Aug. 2011.
- [5] J. Gong, S. Zhou, and Z. Niu, "Optimal power allocation for energy harvesting and power grid coexisting wireless communication systems," *IEEE Transactions on Communications*, vol. 61, no. 7, pp. 3040–3049, Jul. 2013.
- [6] T. Han and N. Ansari, "On optimizing green energy utilization for cellular networks with hybrid energy supplies," *Wireless Communications, IEEE Transactions on*, vol. 12, no. 8, pp. 3872–3882, Aug. 2013.
- [7] J. Gong, J. Thompson, S. Zhou, and Z. Niu, "Base station sleeping and resource allocation in renewable energy powered cellular networks," *IEEE Transactions on Communications*, vol. 62, no. 11, pp. 3801–3813, Nov. 2014.
- [8] G. Piro, M. Miozzo, G. Forte, N. Baldo, L. Grieco, G. Boggia, and P. Dini, "HetNets powered by renewable energy sources: Sustainable next-generation cellular networks," *IEEE Internet Computing*, vol. 17, no. 1, pp. 32–39, Jan. 2013.
- [9] Y. Mao, Y. Luo, J. Zhang, and K. Letaief, "Energy harvesting small cell networks: feasibility, deployment, and operation," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 94–101, Jun. 2015.
- [10] I. Podnar, M. Hauswirth, and M. Jazayeri, "Mobile push: Delivering content to mobile users," in *Proc. 22nd International Conference on Distributed Computing Systems Workshops*, 2002.
- [11] P. Nicosopolitidis, G. I. Papadimitriou, and A. S. Pomportsis, "Using learning automata for adaptive push-based data broadcasting in asymmetric wireless environments," *IEEE Transactions on Vehicular Technology*, vol. 51, no. 6, pp. 1652–1660, Nov. 2002.
- [12] K. Wang, Z. Chen, and H. Liu, "Push-based wireless converged networks for massive multimedia content delivery," *IEEE Transactions on Wireless Communications*, vol. 13, no. 5, pp. 2894–2905, May 2014.
- [13] N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching and device-to-device collaboration: A new architecture for wireless video distribution," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 142–149, Apr. 2013.
- [14] K. Shanmugam, N. Golrezaei, A. F. Molisch, A. G. Dimakis, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.
- [15] K. Poularakis, G. Iosifidis, V. Sourlas, and L. Tassiulas, "Multicast-aware caching for small cell networks," in *IEEE Wireless Communications and Networking Conference (WCNC)*, Apr. 2014.
- [16] P. Blasco and D. Gunduz, "Learning-based optimization of cache content in a small cell base station," in *Communications (ICC), 2014 IEEE International Conference on*, Jun. 2014.
- [17] E. Bastug, M. Bennis, and M. Debbah, "Cache-enabled small cell networks: Modeling and tradeoffs," in *11th International Symposium on Wireless Communications Systems (ISWCS)*, Aug. 2014.
- [18] X. Wang, Y. Bao, X. Liu, and Z. Niu, "On the design of relay caching in cellular networks for energy efficiency," in *IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Apr. 2011.
- [19] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5G wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

- [20] S. Sadr and S. Valentin, "Anticipatory buffer control and resource allocation for wireless video streaming," *arXiv preprint arXiv:1304.3056*, 2013.
- [21] M. Gregori, J. Gomez-Vilardebo, J. Matamoros, and D. Gunduz, "Joint transmission and caching policy design for energy minimization in the wireless backhaul link," in *IEEE Intl Symposium on Information Theory (ISIT)*, Jun. 2015.
- [22] G. Lee, W. Saad, M. Bennis, A. Mehdodniya, and F. Adachi, "Online ski rental for scheduling self-powered, energy harvesting small base stations," in *IEEE International Conference on Communication (accepted)*, 2016.
- [23] Y. Xiao, D. Niyato, Z. Han, and L. A. DaSilva, "Dynamic energy trading for energy harvesting communication networks: A stochastic energy trading game," *IEEE Journal on Selected Areas in Communications*, vol. 33, no. 12, pp. 2718–2734, 2015.
- [24] D. T. Hoang, D. Niyato, P. Wang, and D. I. Kim, "Opportunistic channel access and RF energy harvesting in cognitive radio networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 11, pp. 2039–2052, 2014.
- [25] S. Zhou, J. Gong, Z. Zhou, W. Chen, and Z. Niu, "Greendelivery: Proactive content caching and push with energy-harvesting-based small cells," *IEEE Communications Magazine*, vol. 53, no. 4, pp. 142–149, Apr. 2015.
- [26] M. Cha, H. Kwak, P. Rodriguez, Y.-Y. Ahn, and S. Moon, "I tube, you tube, everybody tubes: Analyzing the world's largest user generated content video system," in *Proc. 7th ACM SIGCOMM Conference on Internet measurement*. ACM, 2007.
- [27] Z. Niu, L. Long, J. Song, and C. Pan, "A new paradigm for mobile multimedia broadcasting based on integrated communication and broadcast networks," *IEEE Communications Magazine*, vol. 46, no. 7, pp. 126–132, Jul. 2008.
- [28] J. Liu, W. Chen, Y. J. Zhang, and Z. Cao, "A utility maximization framework for fair and efficient multicasting in multicarrier wireless cellular networks," *IEEE/ACM Transactions on Networking*, vol. 21, no. 1, pp. 110–120, Feb. 2013.
- [29] B. Zhou, Y. Cui, and M. Tao, "Stochastic content-centric multicast scheduling for cache-enabled heterogeneous cellular networks," *arXiv preprint arXiv:1509.06611*, 2015.
- [30] D. P. Bertsekas, *Dynamic programming and optimal control*, 3rd edition. Athena Scientific Belmont, MA, 2005.
- [31] N. Sharma, D. Krishnappa, D. Irwin, M. Zink, and P. Shenoy, "Greencache: Augmenting off-the-grid cellular towers with multimedia caches," in *Proc. ACM MMSys13*, Feb. 2013.
- [32] A. Kumar and W. Saad, "On the tradeoff between energy harvesting and caching in wireless networks," in *IEEE International Conference on Communication Workshop*, 2015, pp. 1976–1981.
- [33] J. Gong, S. Zhou, Z. Zhou, and Z. Niu, "Proactive push with energy harvesting based small cells in heterogeneous networks," in *IEEE ICC'15, London, UK*, Jun. 2015.
- [34] G. Paschos, E. Baştuğ, I. Land, G. Caire, and M. Debbah, "Wireless caching: Technical misconceptions and business barriers," *IEEE Communications Magazine*, vol. 54, no. 8, pp. 16–22, Aug. 2016.
- [35] L. Richard and J. Burden, *Numerical Analysis*. PWS Publishing Co., Boston, MA, 1985.
- [36] O. Ozel and S. Ulukus, "Achieving AWGN capacity under stochastic energy harvesting," *IEEE Transactions on Information Theory*, vol. 58, no. 10, pp. 6471–6483, 2012.
- [37] L. Kleinrock, *Queueing systems, volume 1: theory*. John Wiley & Sons New York, 1975.